

**In the Claims**

Applicant has submitted a new complete claim set showing amended claims with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing.

Please amend pending claims 1, 5 and 9 as noted below.

1. (Currently amended) A method of preparing an executable program from a plurality of object code modules, each module containing sets of section data and associated ~~relocations~~ relocation instructions, and at least one of said modules further including a macro section containing code sequences at least some of which are likely to be repeatedly included in the executable program and macro relocations associated with said macro section, wherein at least one of said sets of section data includes at least one insertion location where ~~said at least~~ some code sequences are to be inserted and ~~its wherein the~~ associated relocation instructions include a macro call relocation (R\_CALL\_MACRO) identifying a location in the macro section, the method comprising at link time when said executable program is prepared:

reading said sets of section data and relocation instructions;  
on locating said macro call relocation identifying the location in the macro section; and  
inserting said at least some code sequences from said location in the macro section into the set of section data at the insertion location, said at least some code sequences being selected by reading the macro relocations.

2. (Original) A method according to claim 1, wherein said macro relocations calculate conditions resolvable at link time to determine which of said code sequences is to be included in the executable program

3. (Original) A method according to claim 1, wherein said relocations include a relocation which supplies at least one parameter together with an index for holding said parameter in association with the index in a parameter array from which the parameter can be recalled at link time.

4. (Original) A method according to claim 3, wherein said relocations include a relocation which supplies an index for recalling said at least one parameter from the parameter array.

5. (Currently amended) A linker for preparing an executable program for a plurality of object code modules, each module containing sets of section data and ~~relocations~~ associated relocation instructions, at least one of said modules further including a macro section containing code sequences at least some of which are likely to be repeatedly included in the executable program and macro relocations associated with said macro section, wherein at least one of said sets of section data includes at least one insertion location where ~~said at least some~~ code sequences are to be inserted, and ~~its wherein the~~ associated ~~relocations~~ relocation instructions include a macro call relocation identifying a location in the macro section, the linker comprising:

a relocation module for reading the ~~relocations~~ relocation instructions, the relocation module being operable to identify a macro call relocation and to locate said location in the macro section;

a section data module for holding section data to which the ~~relocations~~ relocation instructions relate and arranged to receive said at least some code sequences from the location in the macro section to be inserted at the insertion location; and

a program preparing means which prepares said executable program including said set of section data with the inserted code sequences.

6. (Original) A linker according to claim 5, which comprises a parameter array for holding a set of parameters used by the macro relocation of the macro section each associated with a respective index.

7. (Original) A linker according to claim 5, which comprises a condition array which holds respective values with associated indexes, said indexes being conveyed by the relocation instructions for recalling said values, said values being used to conditionally select said at least some code sequences at link time.

8. (Previously Presented) A method of assembling an object code module for linking to form an executable program, the method comprising:

executing a set of assembler directives including a macro call directive, and responsive to the macro call directive:

naming a location in a macro section in the object code module containing a plurality of code sequences, at least some of which are likely to be repeatedly included in the executable program;

marking at an insertion location IL in a set of section data in the object code module where at least some of said code sequences are to be inserted in the final executable program;

generating in association with the section data a macro call relocation identifying the named location in the macro section; and

generating a set of macro relocations associated with said macro section for selecting said at least some code sequences for insertion at the insertion location.

9. (Currently amended) A computer program product in the form of an object code module which contains sets of section data and ~~relocations~~associated relocation instructions, the module further including a macro section containing code sequences at least some of which are likely to be repeatedly included in the executable program and a set of macro relocations associated with said macro section wherein at least one of said sets of section data includes an insertion location where ~~said at least some~~ code sequences are to be inserted, and ~~its wherein the~~ associated ~~relocations~~ relocation instructions include a macro call relocation identifying a location in the macro section, the computer program product being cooperable with a linker to cause execution of relocation operations by the linker in dependence on said relocations and including identifying the location in the macro section and inserting said at least some code sequences from that location in the macro section in the set of section data at the insertion location.